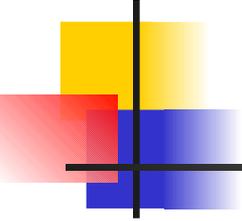


BIF703

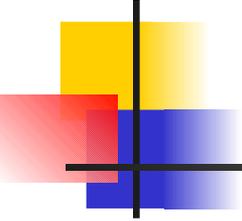
Managing Processes



Agenda

Managing Processes (Jobs)

- Command Grouping
- Running jobs in background (bg)
- Bringing jobs to foreground (fg),
- Background job status (jobs)
- Suspending jobs (CTRL Z)
- Restarting jobs in background
- Displaying/Killing processes or jobs

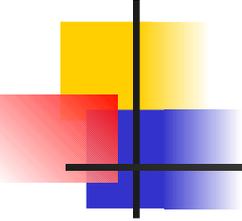


Controlling Processes

A process in UNIX simply represents the running of a command

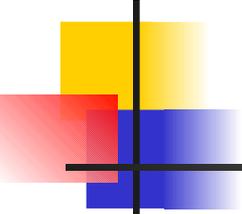
The process structure in UNIX is similar to the hierarchical structure of directories:

- The beginning process is **root**
- **Parent processes** can create **child processes**
- Processes can spawn other processes
- When a **child process** is finished, execution returns to its parent process



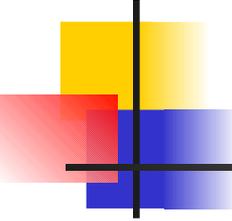
Controlling Processes

- It is important to learn how UNIX processes work, since administering a UNIX system requires knowledge of how to start, stop and monitor processes.
- The UNIX system is built on a series of processes and sub-processes. If you don't understand processes, you could accidentally shut down the UNIX system.
(not a very smart or popular thing to do!)



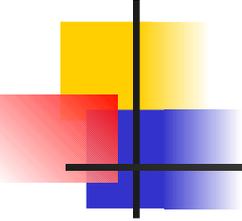
Processes

- For your account, the basic process is the shell (although for the entire system - from root downwards there are many processes).
- When you type in a UNIX command and press ENTER, a child process is created to allow execution of command. During this time the parent process “sleeps” and resumes when child process is completed.
- There is even a command called sleep to suspend the process for a specific number of seconds!



Process Identification (PID)

- Process Identification Numbers (PIDs) are used to keep track of separate processes
- To view a compact listing of process ID numbers, you can enter `ps`
- To view a detailed listing of process ID numbers (relating child (PID) & parent processes (PPID)) enter `ps -l`
- To view listing of processes for the entire system, enter `ps -ef`

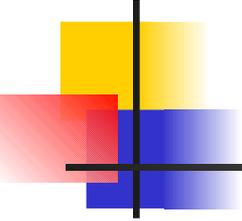


Command Grouping

You can run a set of commands on the same command line by typing such commands as:

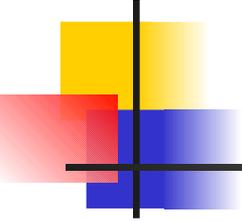
```
ls ; who ; date <ENTER>
```

- You can also, combine commands or files to be run in both the foreground or background (eg `ls ; who` or `a & b ; c`)



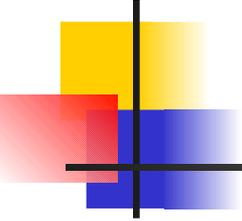
Command Grouping

- The UNIX OS has the ability to run programs in the foreground and in the background. This feature allows users of the UNIX system to run lengthy processes without interfering with their other tasks.
- To run a command or script in the background, type the symbol `&` after the command or filename (eg. `who &`). Note that spaces are allowed...



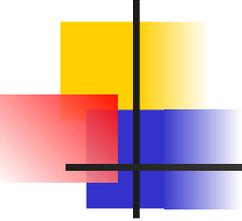
Command Grouping

- It is important to note that although these commands appear to be in a sequence, the shell may execute these in a different order may schedule jobs in a order
- For example, you may want to run both commands a & b in the background, and run command c in the foreground. Here is the command:
- `(a:b)&c`



Bringing processes to foreground

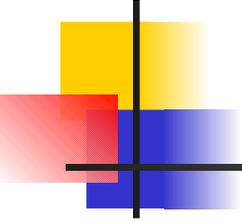
- When processes are run in the foreground, the shell will wait for the command to finish before allowing the user to enter another command (thus background processes are useful)
- If a process or processes are in the background (or suspended), you can bring it to the foreground by typing **fg**



Displaying job status (jobs command)

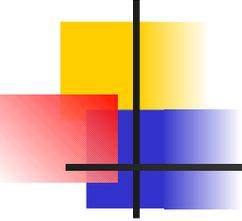
You can display the status of jobs that are running in the background by typing the UNIX command `jobs`

- jobs commands give a listing of jobs running in the background:
- jobs with a plus sign “+” indicates default job to bring to foreground by entering fg
- Can bring numbered job to foreground by entering `fg %(job#)` - Note space between fg & %



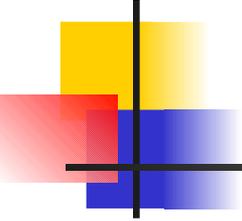
Suspending jobs (CTRL Z)

- You can suspend or stop the execution of processes while they are running in the “foreground” by pressing `<CTRL><Z>`
- When you press these keys, the process is suspended and placed into the background in order to free up the shell for other operations.
- You can resume process by bringing it to the foreground or have it run in background



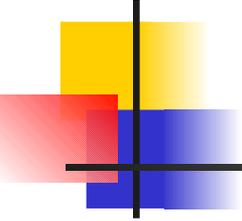
Restarting Suspended Processes in the Background

- A process may take such a long time that it is better to suspend the job (i.e. send it to the background, and then have it run in the background). This will leave you to operate in the foreground
- If a process or processes are suspended in the background, you can have it continue running in the background by entering `bg` (rules such as `bg %(job#)` also apply)



Terminating processes or jobs (kill)

- Depending on your permissions, you may be able to abort or “kill” a process.
- Only root is allowed to kill other user’s processes, but as a user, you can processes that you have created.
- Killing a process is also useful to halt processes that are running in the background since <CTRL><C> or DELETE keys in foreground won’t work



Procedure to “Kill” a Process

Procedure:

- We are assuming that you are killing one of your own process (you are logged in...)
- View process by typing `ps -l`
- Carefully locate PID to kill
- Type `kill %(job#)` to kill that job number (can also use `kill PID` or `kill -9 PID` to kill process. The `kill` command by itself kills more recent stopped job